

In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

- 1 1. (Currently Amended) The structure of a subpipelined
2 translation embodiment providing binary compatibility between a
3 base architecture and migrant architecture of a VLIW architecture
4 comprising:
5 a VLIW architecture comprising a base architecture and a
6 migrant architecture and having a base execution mode and a migrant
7 execution mode;
8 an instruction fetch unit for simultaneously fetching from
9 memory a group of a plurality of instructions, each such group
10 forming a fetch packet, said instruction fetch unit assigning each
11 fetch packet an operating mode in dependence upon the execution
12 mode at the time the request was made to the memory for the fetch
13 packet;
14 a shared datapath by both the base and migrant architectures
15 for parsing said base architecture mode and migrant architecture
16 mode fetch packets into execute packets of instructions within said
17 fetch packet that can be executed simultaneously;
18 a base architecture control circuit for dispatching execute
19 packet instructions having a base execution mode;
20 a migrant architecture control circuit for dispatching execute
21 packet instructions having a migrant execution mode;
22 a base architecture decode connected to said shared datapath
23 and said base architecture control circuit for decoding an execute
24 packet in said base mode and generating a corresponding machine
25 word;
26 a migrant architecture decode connected to said shared
27 datapath and said migrant architecture control circuit for decoding

28 an execute packet in said migrant mode and generating a
29 corresponding machine word;

30 a multiplexer having at least ~~two~~ three inputs and one machine
31 word output wherein ~~one~~ a first input is the machine word output of
32 said migrant architecture decode, a second ~~and the other~~ input is
33 the machine word output of said base architecture decode and a
34 third input receiving a no operation instruction machine word, said
35 multiplexer choosing in dependence upon the operating mode of said
36 fetch packet; and

37 execute hardware connected to said multiplexer for executing
38 execute packet instructions on execution units corresponding to
39 said machine word chosen by said multiplexer.

2. (Canceled)

1 3. (Original) The structure according to Claim 1, wherein
2 said machine word also controls registers.

1 4. (Original) The structure according to Claim 1, wherein said
2 machine word controls a global register file, which supplies
3 operands to all hardware execution units and accepts results of all
4 hardware execution units.

1 5. (Original) The structure according to Claim 4, wherein
2 said machine word controls local register files that supply
3 operands to either local execution hardware functional units or
4 neighbor hardware execution functional units subsequent to said
5 machine word controlling said global register file.

1 6. (Original) The structure according Claim 5, wherein said
2 machine word controls the various types of execution hardware that
3 evaluate functions on the operands to produce the results of said

4 hardware execution units subsequent to said machine word
5 controlling said local register files.

1 7. (Original) The structure according to Claim 1, wherein the
2 base and migrant architecture decode units translates opcodes to
3 the control signals required to execute the specified instructions
4 on the execution hardware functional units.

1 8. (Previously Presented) The structure according to claim 1,
2 and further comprising said migrant architecture control circuit
3 for issuing no-operation instruction to preserve the semantics of
4 the instruction in the migrant architecture due to differences in
5 instruction latencies between the base architecture and the migrant
6 architecture.

1 9. (Original) The structure according to Claim 1, wherein
2 said VLIW architecture is a Digital signal Processor (DSP).

1 10. (Currently Amended) A method of providing binary
2 compatibility between a base architecture and migrant architecture
3 of a VLIW architecture comprising the steps of:
4 simultaneously fetching from a memory a group of a plurality
5 of instructions, each such group forming a fetch packet;
6 assigning each fetch packet an operating mode in dependence
7 upon the execution mode at the time the request was made to the
8 memory for the fetch packet;
9 parsing said base architecture mode and migrant architecture
10 mode fetch packets into execute packets of instructions within said
11 fetch packet that can be executed simultaneously;
12 dispatching execute packet instructions having a base
13 execution mode;

14 dispatching execute packet instructions having a migrant
15 execution mode;
16 decoding an execute packet in said base mode and generating a
17 corresponding machine word;
18 decoding an execute packet in said migrant mode and generating
19 a corresponding machine word;
20 choosing one machine word output, in dependence upon the
21 operating mode of said fetch packet, between the machine word
22 decoded in said migrant mode, ~~and~~ the machine word decoded in said
23 base mode and a no operation machine word;
24 controlling the execution hardware units with said chosen
25 machine word.

11. (Canceled)

1 12. (Original) The method according to Claim 10, and further
2 comprising controlling registers with said machine word.

1 13. (Original) The method according to Claim 10, and further
2 comprising controlling a global register file with said machine
3 word, which supplies operands to all hardware execution units and
4 accepts results of all hardware execution units.

1 14. (Original) The structure according to Claim 13, and
2 further comprising controlling local register files that supply
3 operands to either local execution hardware functional units or
4 neighbor hardware execution functional units subsequent to said
5 controlling said global register file.

1 15. (Original) The method according Claim 14, and further
2 comprising controlling the various types of execution hardware that
3 evaluate functions on the operands to produce the results of said

4 hardware execution units subsequent to controlling said local
5 register files.

1 16. (Original) The method according to Claim 10, and further
2 comprising translating opcodes to the control signals required to
3 execute the specified instructions on the execution hardware
4 functional units within the base and migrant architecture decode
5 units.

1 17. (Original) The method according to claim 10, wherein said
2 VLIW architecture is a Digital Signal Processor (DSP).

1 18. (Currently Amended) The method according to Claim 10 and
2 further comprising the step of issuing no-operation instruction
3 ~~from said migrant architecture control circuit,~~ to preserve the
4 semantics of the instructions in the migrant architecture due to
5 differences in instruction latencies between the base architecture
6 and the migrant architecture.

1 19. (New) The structure of a subpipelined translation
2 embodiment providing binary compatibility between a base
3 architecture and migrant architecture of a VLIW architecture
4 comprising:
5 a VLIW architecture comprising a base architecture and a
6 migrant architecture and having a base execution mode and a migrant
7 execution mode;
8 an instruction fetch unit for simultaneously fetching from
9 memory a group of a plurality of instructions, each such group
10 forming a fetch packet, said instruction fetch unit assigning each
11 fetch packet having an operating mode in dependence upon the
12 execution mode at the time the request was made to the memory for
13 the fetch packet;

14 a shared datapath by both the base and migrant architectures
15 for parsing said base architecture mode and migrant architecture
16 mode fetch packets into execute packets of instructions within said
17 fetch packet that can be executed simultaneously;

18 a base architecture control circuit for dispatching execute
19 packet instructions having a base execution mode;

20 a migrant architecture control circuit for dispatching execute
21 packet instructions having a migrant execution mode, said migrant
22 architecture control circuit issuing a no-operation instruction to
23 preserve the semantics of the instruction in the migrant
24 architecture due to differences in instruction latencies between
25 the base architecture and the migrant architecture;

26 a base architecture decode connected to said shared datapath
27 and said base architecture control circuit for decoding an execute
28 packet in said base mode and generating a corresponding machine
29 word;

30 a migrant architecture decode connected to said shared
31 datapath and said migrant architecture control circuit for decoding
32 an execute packet in said migrant mode and generating a
33 corresponding machine word;

34 a multiplexer having at least two inputs and one machine word
35 output wherein a first input is the machine word output of said
36 migrant architecture decode and a second input is the machine word
37 output of said base architecture decode, said multiplexer choosing
38 in dependence upon the operating mode of said fetch packet; and

39 execute hardware connected to said multiplexer for executing
40 execute packet instructions on execution units corresponding to
41 said machine word chosen by said multiplexer.

1 20. (New) The structure according to Claim 19, wherein said
2 machine word also controls registers.

1 21. (New) The structure according to Claim 19, wherein said
2 machine word controls a global register file, which supplies
3 operands to all hardware execution units and accepts results of all
4 hardware execution units.

1 22. (New) The structure according to Claim 21, wherein said
2 machine word controls local register files that supply operands to
3 either local execution hardware functional units or neighbor
4 hardware execution functional units subsequent to said machine word
5 controlling said global register file.

1 23. (New) The structure according Claim 22, wherein said
2 machine word controls the various types of execution hardware that
3 evaluate functions on the operands to produce the results of said
4 hardware execution units subsequent to said machine word
5 controlling said local register files.

1 24. (New) The structure according to Claim 19, wherein the
2 base and migrant architecture decode units translates opcodes to
3 the control signals required to execute the specified instructions
4 on the execution hardware functional units.

1 25. (New) The structure according to Claim 19, wherein said
2 VLIW architecture is a Digital signal Processor (DSP).

1 26. (New) A method of providing binary compatibility between
2 a base architecture and migrant architecture of a VLIW architecture
3 comprising the steps of:
4 simultaneously fetching from a memory a group of a plurality
5 of instructions, each such group forming a fetch packet;

6 assigning each fetch packet an operating mode in dependence
7 upon the execution mode at the time the request was made to the
8 memory for the fetch packet;
9 parsing said base architecture mode and migrant architecture
10 mode fetch packets into execute packets of instructions within said
11 fetch packet that can be executed simultaneously;
12 dispatching execute packet instructions having a base
13 execution mode;
14 dispatching execute packet instructions having a migrant
15 execution mode;
16 decoding an execute packet in said base mode and generating a
17 corresponding machine word;
18 decoding an execute packet in said migrant mode and generating
19 a corresponding machine word;
20 issuing a no-operation instruction to preserve the semantics
21 of the instructions in the migrant architecture due to differences
22 in instruction latencies between the base architecture and the
23 migrant architecture;
24 choosing one machine word output, in dependence upon the
25 operating mode of said fetch packet, between the machine word
26 decoded in said migrant mode the machine word decoded in said base
27 mode and the no-operation instruction issued to preserve the
28 semantics of the instructions in the migrant architecture;
29 controlling the execution hardware units with said chosen
30 machine word.

1 27. (New) The method according to Claim 26, and further
2 comprising controlling registers with said machine word.

1 28. (New) The method according to Claim 26, and further
2 comprising controlling a global register file with said machine

3 word, which supplies operands to all hardware execution units and
4 accepts results of all hardware execution units.

1 29. (New) The structure according to Claim 28, and further
2 comprising controlling local register files that supply operands to
3 either local execution hardware functional units or neighbor
4 hardware execution functional units subsequent to said controlling
5 said global register file.

1 30. (New) The method according Claim 29, and further
2 comprising controlling the various types of execution hardware that
3 evaluate functions on the operands to produce the results of said
4 hardware execution units subsequent to controlling said local
5 register files.

1 31. (New) The method according to Claim 26, and further
2 comprising translating opcodes to the control signals required to
3 execute the specified instructions on the execution hardware
4 functional units within the base and migrant architecture decode
5 units.

1 32. (New) The method according to claim 26, wherein said VLIW
2 architecture is a Digital Signal Processor (DSP).